

拒絶査定

特許出願の番号	特願2006-521512
起案日	平成21年 8月 7日
特許庁審査官	岩橋 龍太郎 3790 5E00
発明の名称	グリッドプラウザコンポーネント
特許出願人	エスアーペー アーグー
代理人	谷 義一 (外 1名)

この出願については、平成21年 4月10日付け拒絶理由通知書に記載した理由Aによって、拒絶をすべきものです。

なお、意見書及び手続補正書の内容を検討しましたが、拒絶理由を覆すに足りる根拠が見いだせません。

備考

・請求項1～3について

出願人は平成21年7月13日付けで手続補正書を提出するとともに、意見書において、本願発明は、「エッジおよび頂点を有するグラフを備え、前記頂点は、グリッドノードを表し、前記エッジは、グリッドコンピューティングネットワークにおける2つのグリッドノードの関連づけを表すことを特徴とするグラフィカルユーザインターフェイス（GUI）を表示し、前記グラフィカルユーザインターフェースがユーザのクリックを受け取ると、前記ノードによって表される、コンピュータ上で実行されるコンピュータグリッドアプリケーションが、各エッジについて、前記ユーザに見せられることを特徴とするプログラム」であるのに対して、特開2000-078156号公報（以下、「引用文献1」という。）には、その旨の記載がないので、本願発明は、引用文献1に記載された発明に基づいて、当業者が容易に発明をすることができたものではない旨主張している。

しかしながら、平成21年4月10日付け拒絶理由通知書に示したとおり、引用文献1には、機器をノードで表し、ノード間を線で表すことが記載されており、また、グリッドコンピューティングネットワークは、文献を示すまでもなく周知であるから、引用文献1に記載の発明を、周知のグリッドコンピューティングネットワークに適用して、エッジおよび頂点を有するグラフを備え、前記頂点は、グリッドノードを表し、前記エッジは、グリッドコンピューティングネットワークにおける2つのグリッドノードの関連づけを表すグラフィカルユーザインターフェイス（GUI）を表示することに、困難性はない。

そして、コンピューティングネットワークのGUITにおいて、ユーザの操作を受け取ると、ノードによって表されるコンピュータ上で実行されるジョブ等の状況を、前記ユーザに見せられるようにすることも、周知の技術であり、例えば、特開平07-311743号公報（特に、段落【0017】、【図10】、【図11】を参照。）などに記載されている。

したがって、引用文献1に記載の発明に周知技術を適用して、本願の上記請求項に係る発明を構成することは、当業者が容易に想到し得るものである。

よって、出願人の上記主張は採用されない。

この査定に不服があるときは、この査定の謄本の送達があった日から3月以内（在外者にあっては、4月以内）に、特許庁長官に対して、審判を請求することができます（特許法第121条第1項）。

（行政事件訴訟法第46条第2項に基づく教示）

この査定に対しては、この査定についての審判請求に対する審決に対してのみ取消訴訟を提起することができます（特許法第178条第6項）。

上記はファイルに記録されている事項と相違ないことを認証する。

認証日 平成21年 8月10日 経済産業事務官 吉越 誠

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-311743
 (43)Date of publication of application : 28.11.1995

(51)Int.Cl.

G06F 15/16
G06F 15/16

(21)Application number : 06-102451

(71)Applicant : HITACHI LTD

(22)Date of filing : 17.05.1994

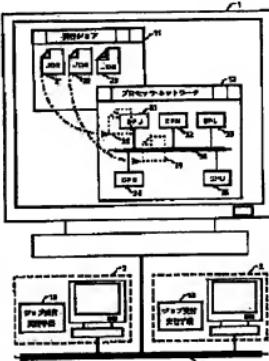
(72)Inventor : KAGEYAMA KEIJI
KO YOSHIMITSU
TSURUKI MASAKI

(54) PROCESSOR SPECIFICATION SYSTEM

(57)Abstract:

PURPOSE: To enable a user to instruct a computer system to execute the distributed precesssing and execute the distributed processing interactively by instructing the computer system to processing specified processing file on a specific device.

CONSTITUTION: A job management means 11 regards one selected processor in a network as a server machine 2 and defines the execution of a job, described in a previously selected job file, on the server computer 2. An icon is selected through operation specifying the job file, processor, or network in the depression order of the button of a painter device or operation specifying the job file where the button is pressed and the processor or network where the button is released. When they are selected where the button is pressed and released, the previously selected icon is displayed as a job moving icon 25 following the pointer device while the pointer device is moved.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-311743

(43)公開日 平成7年(1995)11月28日

(51)Int.Cl.^{*}
G 0 6 F 15/18識別記号
G 0 6 F 15/16
3 7 0

F I

技術表示箇所

G 0 6 F 15/16 4 2 0 J

審査請求 未請求 請求項の数7 O.L. (全11頁)

(21)出願番号 特願平6-102451
(22)出願日 平成6年(1994)5月17日

(71)出願人 000005108
 株式会社日立製作所
 東京都千代田区神田駿河台四丁目6番地

(72)発明者 影山 啓二
 実業団土浦市神立町502番地 株式会社日
 立製作所機械研究所内

(72)発明者 廣 喜光
 神奈川県川崎市幸区麻生890 株式会社
 日立製作所情報システム事業部内

(72)発明者 鶴来 昌樹
 実業団土浦市神立町502番地 株式会社日
 立製作所機械研究所内

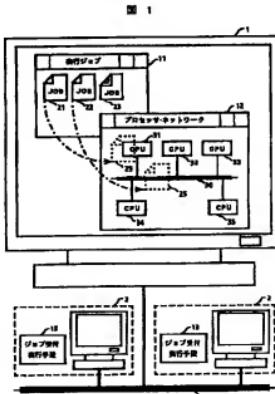
(74)代理人 弁理士 小川 駿男

(54)【発明の名称】 プロセッサ指定方式

(57)【要約】

【構成】プロセッサ指定方式は、ユーザ計算機1、ユーザ計算機1で動作し、ユーザジョブやプロセッサなどをグラフィックイメージで表示・管理するジョブ管理手段1 1、プロセッサ管理手段1 2、ネットワーク3に接続されているサーバ計算機2、サーバ計算機2で動作し、ジョブの受付・実行を行うジョブ受付実行手段1 3より構成されている。

【効果】ユーザジョブやプロセッサなどをグラフィックイメージで表示・管理することにより、分散処理や並列処理の実行指示を対話的に行うことができ、かつ、状態を視覚的に捉えることが可能となる。



【特許請求の範囲】

【請求項1】ユーザインターフェイスを支援するためのワインドウシステムが稼働し、一つ以上の処理プロセッサを持つユーザ計算機と、前記ユーザ計算機の他に、前記ユーザ計算機とネットワークで接続されている一つ以上の処理プロセッサを持つ1台以上のサーバ計算機と、前記ユーザ計算機上で処理内容を記述した処理ファイルを、マウスなどのポイントターピスで特定可能なグラフィックイメージ、または、アイコンで操作できる処理ファイル管理手段と、前記ユーザ計算機、前記サーバ計算機、前記ネットワークをアイコン、またはこれを含むグラフィックイメージで表示し、ポイントターピスの指示位置から指示対象を特定できるプロセッサ管理手段となる計算機システムにおいて、前記ポイントターピスを用いて前記処理ファイル管理手段が管理している処理ファイルの一つを特定し、次に前記ポイントターピスを用いて前記特定処理ファイルのアイコンを移動し、前記プロセッサ管理手段が管理している装置を特定することにより、前記特定処理ファイルが特定装置上で処理されることを前記計算機システムに指示できることを特徴とするプロセッサ指定方式。

【前記項2】請求項1において、複数の処理単位が記述されている処理ファイルを特定した場合、または、ポイントターピスの領域描画動作、異なるボタンの押下、ボタンの押下回数などにより、一つまたは複数の処理ファイルを同時に特定した場合、前記処理ファイルに記述されている処理単位をそれぞれ独立なジョブとして認識する機能を備えた処理ファイル管理手段、および、ポイントターピスの領域描画動作、異なるボタンの押下、ボタンの押下回数などにより、一つまたは複数の処理装置を同時に特定する機能、処理装置として特定された処理プロセッサがマルチプロセッサ、並列計算機などのように複数の独立動作可能なCPUを有している場合、プロセッサが特定されるとときに複数のCPUをそれぞれ独立なプロセッサとして認識する機能を有し、かつ、ユーザの設定により一つのプロセッサとして扱うモードとの切替えが可能である機能を備えたプロセッサ管理手段を有するプロセッサ指定方式。

【請求項3】請求項1において、処理装置にネットワークを表す图形、アイコンが指定された場合、前記ネットワークに接続されている処理装置すべてを指定されたものとするプロセッサ指定方式。

【請求項4】請求項1、2または3において、特定された処理装置が複数の場合、処理対象として認識された処理単位が一つの場合には特定された処理装置の中から一つに、処理単位が複数の場合には個々の処理単位をユーザの指定があればそのとおりに、なければ処理単位の内容、または、処理装置の処理性能に応じて、処理装置の割当を行い、処理装置の割当が確定した処理ファイルから順に処理ファイルのアイコンを割当られた処理

装置のアイコンまで動かすことにより、システムが処理装置を割当てたことを視覚的にユーザに示す機能を有するプロセッサ指定方式。

【請求項5】請求項1、2、3または4において、処理ファイルが、ユーザに特定された、または、システムにより割当てられた処理装置で実行されている時、処理ファイルアイコンやプロセッサアイコンの表示を変化させることにより、前記ジョブが前記プロセッサで実行中であることをユーザが視覚的に認識できるプロセッサ指定方式。

【請求項6】請求項1、2、3または4において、ポイントターピスを用いて前記処理単位管理手段が管理している処理ファイルの一つを特定し、次に前記ポイントターピスを用いて前記特定処理ファイルのアイコンを移動することなく、前記プロセッサ管理手段が管理している装置を特定することにより、前記特定処理単位が前記特定装置上で処理されることを計算機システムに指示できるプロセッサ指定方式。

【請求項7】請求項1において、前記プロセッサ管理手段が管理している装置の性能や状態を、色、形、移動率、実効性能値、などの表示により、処理装置の特定時に装置選択の判断材料をユーザに提示する機能を有するプロセッサ指定方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、独立に動作可能な複数のプロセッサを利用して分散処理を行うことが可能な計算機システムにおいて、ユーザが対話的に処理対象のプロセッサへの割当を行える手段を設けることにより、計算機システムに不慣れなユーザでも容易に分散処理を行えるプロセッサ指定方式に関する。

【0002】

【従来の技術】複数のプロセッサで分散処理する際のプロセッサの割当方法は、特開平3-85664号：輻田和範（日本電気株式会社）、特開平4-34640号：中西浩一（日本電気株式会社）、特開平4-223547号：沢部光太郎（川崎製鉄株式会社）などにあるように、すべて計算機システムが与えられたジョブについて適切だと思われるプロセッサを判断して割当てる方式のみであった。割当での判断は、各プロセッサの処理性能、一定時間毎に検出する負荷レベルなどの状態情報、ジョブの内容や処理量などのジョブ特性情報をもとに、指定した処理時間が最も短いものを最適なプロセッサとすることにより行われる。

【0003】実行プロセッサを直接指定するには、NFS（（米）Sun Microsystems社の登録商標）のユーティリティにあるonコマンドが利用可能である。ただし、計算機システムによってはこのコマンドがない場合もあり、また、コマンド形式でジョブの実行依頼を発行できない場合には、このコマンドを利用できない。

【0004】

【発明が解決しようとする課題】従来技術にあるようなプロセッサ選択方式では、個々のジョブに対してどのプロセッサが最適であるかを計算機システム側の処理を行うことにより決定する。この処理は、各プロセッサの処理性能、負荷状態などの情報、ジョブの内容や処理量などのジョブ特性情報から、すべてのプロセッサについて、ジョブの実行処理に要する時間を推定し、最も短い時間となるものを最適なプロセッサとしてジョブの割当てを行なう。このとき、各プロセッサの負荷状態の情報入手には、各プロセッサ負荷状態をチェックした後、情報をジョブの割当てを行なうプロセッサに送信するという処理が必要となるため時間がかかる。また、最適なプロセッサを選択するためにはジョブ処理量の高精度な予測が必要となるが、ジョブの種類や、入力データにより処理量が変化することが多く、精度の高い予測には時間を要する。このため、即時実行したい場合には、onコマンドなどを用いて直接ユーザがプロセッサを指定し、ジョブの実行依頼を発行する必要がある。しかし、前述したように、計算機システムによってはこのコマンドがない、また、コマンド形式でジョブの実行依頼を発行できない場合はこのコマンドを用いてできないことがある。さらに、計算機システムに不慣れなユーザでは、コマンド入力形式を使いこなすことが困難である。以上の課題を解決するためにには、従来技術を用ひるプロセッサ自動割当方式、及び、プロセッサの直接指定方式を備え、かつ、これらの指定方式による操作イメージをグラフィカルに行えるプロセッサ指定方式が必要である。

【0005】

【課題を解決するための手段】図1に本発明によるプロセッサ指定方式を示す。本発明のプロセッサ指定方式は、ユーザが入出力処理を行うユーザ計算機1、ユーザ計算機1で動作し、ユーザが処理しようとしている一つ以上のジョブ(図ではJOB1(21)、JOB2(22)、JOB3(23)の三つを例としている)をアイコンシ、または、グラフィックイメージ(以下、アイコンと記す)で表示し、管理するジョブ管理手段11、ユーザ計算機1で動作し、ネットワーク3およびネットワーク3に接続されているサーバ計算機2をアイコンで表示。管理するプロセッサ管理手段12、サーバ計算機2で動作し、ジョブ管理手段11、プロセッサ管理手段12とネットワーク3を通して通信を行い、ジョブの受付・実行、および、自プロセッサの情報送信を行なうジョブ受付手段13より構成されている。

【0006】

【作用】本発明のプロセッサ指定方式では、以下の手順により基本処理を実行する。

【0007】(1) ユーザは、分散処理を行なう際、ユーザ計算機1において、まず、ジョブ管理手段11を起動する。ジョブ管理手段11は管理対象となるジョブア

イルを表示するためのウインドウを生成し、これにジョブファイルをアイコンで表示する。その後、ジョブ管理手段11はプロセッサ管理手段12を起動する。プロセッサ管理手段12は各プロセッサ、ネットワークを表示するためのウインドウを生成し、ネットワークにジョブ受付手段13を起動するための制御コードをブロードキャストする。制御コードを受け付けたサーバ計算機2はジョブ受付手段13を起動し、自プロセッサの負荷状態をチェックした後、その結果をプロセッサ管理手段12へ送信する。サーバ計算機2がシステム管理者により、ジョブ管理手段11の要求に応じないよう規定されている場合は、その情報をプロセッサ管理手段12へ送信する。プロセッサ管理手段12はジョブ受付手段13から送信される情報をもとに、サーバ計算機2の状況の色の変化やパターンにより区別し、サーバ計算機2をアイコンで表示する。

【0008】(2) 上記(1)による準備完了後、ジョブ管理手段11はユーザが操作するポインタデバイスの位置と状態(ボタンが押されている/押されていない、など)から何が選択されるかを識別する。ジョブファイル、プロセッサの順に選択された場合、ジョブ管理手段11は選択されたジョブファイルに記述されているジョブを、選択されたプロセッサに対応するサーバ計算機で実行することを定義する。ジョブファイル、ネットワークの順に選択された場合、ジョブ管理手段11は選択されたネットワーク内の一つのプロセッサをサーバ計算機とし、先に選択されたジョブファイルに記述されているジョブを、サーバ計算機で実行することを定義する。アイコンの選択は、ポインタデバイスのボタンの押下順にジョブファイル、プロセッサまたはネットワークを特定する操作、または、ボタンを押した場所でジョブファイルを、離した場所でプロセッサまたはネットワークを特定する操作、により行われる。ボタンの押した場所、離した場所で選択する場合、ポインタデバイスを移動させている間、先に選択したアイコンをジョブ移動アイコン25としてポインタデバイスに追従するよう表示する。

【0009】(3) ジョブ管理手段11は、ジョブをどのサーバ計算機で実行するかを確定した後、ジョブファイルからジョブのプログラム名、入出力ファイル名などのジョブ情報をとり出し、ネットワークを通じてサーバ計算機のジョブ受付手段13に送信する。ジョブ受付手段13はジョブの実行に必要なプログラム、入力データなどの有無をチェックし、ない場合にはファイル転送、ユーザ計算機のファイルシステムのマウントなどの手段を用いてジョブの実行準備を行う。ジョブ受付手段13はジョブの実行準備完了後、ジョブ管理手段11にジョブ開始メッセージを送信し、ジョブの実行を開始する。ジョブ管理手段11はジョブ開始メッセージを受信すると、ジョブの番号、サーバ計算機名をプロセッサ管理手段12へ送り、プロセッサ管理手段12は

5

アイコンの色やパターンを変えることにより、プロセッサに対応するサーバ計算機でジョブが「実行中であることをユーザーに示す。

【0010】(4) ジョブ受付実行手段13はジョブの実行完了後、ジョブ管理手段11にジョブ終了メッセージを送信し、さらに、ジョブ実行による結果出力ファイルを、ファイル名の送信、ファイル転送などの手段を用いてジョブ管理手段11の管理に移す。ジョブ管理手段11はジョブ終了メッセージを受信、出力ファイルの管理を移されると、出力ファイルをアイコンで表示し、さらに、ジョブ終了の情報をプロセッサ管理手段12へ送る。プロセッサ管理手段12はジョブ終了の情報により、プロセッサのアイコンをもの状態に戻す。

【0011】上記(1)～(4)の処理により、ユーザーが対話的に分散処理の指示、および、分散処理の実行を行うことを可能とする。

【0012】

【実施例】まず、本発明の基本的な実施例を図2ないし図11を用いて説明する。

【0013】図2はジョブの選択方法を示している。ユーザーがジョブ管理手段11を起動すると、ジョブ管理手段11は起動されディレクトリにあるファイルのファイル名、ファイル属性を検索し、ジョブ実行に関するものを専用のウインドウにアイコンで表示する。図2の上の図は、ジョブの実行内容が記述されているファイルが三つあり、そのうち、ジョブアイコン23は複数の実行単位が含まれており、また、サブディレクトリアイコン24があるという例を示したものである。ユーザーは、マウスを操作し、実行しようとするファイルのアイコン、例えばジョブアイコン21へマウスカーソル10を移動し、マウスのボタンを押す。ジョブ受付実行手段11はウインドウシステムを通してマウスのボタンが押されたという情報を取得し、図2の中央の図のように押された場所にあるアイコンをジョブ選択アイコン211に変える。この後、ユーザーがマウスのボタンを押したままマウスを動かすと、ジョブ管理手段11はマウスの位置を一定時間毎に取得し、位置にジョブ移動アイコン25を表示する。このとき、1回前に取得したマウスの位置に表示していたものを削除することにより、動かしていることをユーザーが視覚的に捉えることができるようになる。ものの位置に表示していたジョブ選択アイコン211は、状態が変化したことを示すためにジョブ選択・移動アイコン212に変える。また、図2の下の図のように、ジョブ移動アイコン25は、外形態のみとするなど、できるだけ表示を高速に行えるものにすると、マウスの位置取得の時間間隔が短くてもジョブ移動アイコン25の移動を容易にアニメーションにすることができる。

【0014】次に、プロセッサの指定方法を示す。図3は直接実行プロセッサを指定する方法を示したものであ

る。ジョブ管理手段11はジョブファイルの表示用ウィンドウを生成した後、プロセッサ管理手段12を起動し、プロセッサ管理手段12はプロセッサネットワークをアイコンで表示するウィンドウを生成する。ユーザーはジョブを選択したままマウスを移動させ、プロセッサ管理手段12のウィンドウに移動用アイコン12を移動させ、選択ジョブを実行させたいCPU31へ重ね、マウス移動時に押していたボタンを離す。この操作により、ジョブアイコン21に記述されている処理単位をCPU31で実行したサーバ計算機で実行することを定義する。この操作を示したものであり、例としてCPU31を指定している。ジョブ管理手段11は選択されたアイコンに対するジョブファイルを実行するサーバ計算機が定義された際、ジョブファイルに記述されている処理単位の数をチェックし、処理単位数がサーバ計算機のプロセッサ数よりも大きいときは、一つのプロセッサが複数の処理単位を実行することになることをユーザーに通知する。ユーザーは内容を承認するか、プロセッサの指定を取り消すかを選択する。ここで承認された場合、ジョブ管理手段11はサーバ計算機に対してジョブの実行依頼を行なう。図4はプロセッサを選択する際、ネットワークを示すアイコンを選択した場合を示している。ジョブを選択したままマウスを移動させ、ネットワークのアイコンのところでマウス移動時に押していたボタンを離すと、プロセッサ管理手段12はネットワークに接続されたサーバ計算機全部が選択されたという認識をする。このとき、ネットワークに接続された計算機の内、ジョブ受付実行手段13が動作している計算機のみ選択可能であり、ジョブ受付実行手段13が動作していない計算機は選択対象となるしない。ジョブ管理手段11は選択されたサーバ計算機の中から、ユーザーが選択したジョブファイルを実行するのに適当なサーバ計算機を選択する。このとき、ジョブ管理手段11はプロセッサ管理手段12より、サーバ計算機の処理能力、負荷状況、利用者数などの情報をもらい、ジョブの内容を調べて処理量を推定し、選択されたサーバ計算機群の各のプロセッサについて処理時間を推定し、最も処理時間の短いものを最適なサーバ計算機とみなす。この手順を図5を用いて説明する。サーバ計算機2ではログインユーザ数、CPIの稼働状況を定期的にチェックする。ユーザ数チェック手段121のデータとCPU稼働率チェック手段122のデータより、現処理能力推定手段123が現在の処理能力を推定する。処理能力と、CPU稼働率履歴保持手段124が持つ過去のCPU稼働率を集めた稼働履歴とから、処理能力分布推定手段125が現在以降の処理能力分布を推定する。ユーザ計算機1では、ジョブ内容確認手段111がユーザの指示したジョブの内容をチェックし、処理時間推定手段112がジョブの実行にかかる処理量を推定する。処理時間推定手段113は、サーバ計算機2が推定したサーバ計算機2の処理能力分布を

7

積分し、積分値が処理量と一致するような時間を求める。時間を処理時間とし、これを比較手段114がジョブを実行可能なサーバ計算機2について比較し、最も処理時間の小さいものを最適なサーバ計算機として選択する。ユーザのジョブファイルを実行するサーバ計算機が確定すると、ジョブ管理手段11は図6に示すようにジョブファイルアイコンをサーバ計算機のアイコンへ重ねることにより、ジョブファイルがサーバ計算機で実行されることが決定したことをユーザに示す。このとき、ジョブファイルアイコンをネットワークのケーブルを示す图形に沿って移動させると、ユーザにネットワークを利用している感覚を伝えることができる。

【0015】次に図7を用いてジョブを実行する方法を説明する。ジョブ管理手段11は、選択したジョブを実行するサーバ計算機が確定すると、サーバ計算機のジョブ受付実行手段13に選択したジョブアイコンの実体であるジョブファイル50を参照し、ジョブのプログラム名や入出力ファイルのパス名を取り出す。ここで、入力データファイルが実際に存在するかどうかの確認を行い、実在している場合は、ネットワーク3を介してプログラム名などをサーバ計算機のジョブ受付実行手段13に送信する。ジョブ受付実行手段13は入力データファイル51のパス名をチェックし、自プロセッサが入力データファイル51を直接アクセスできない場合には、ファイル転送などにより入力データファイル51のコピーをローカル入力データファイル61として作成する。この後、ジョブ受付実行手段13はジョブを実行するためのプロセスを起動する。起動された実行ジョブ14は入力データファイル51、または、ローカル入力データファイル61を入力データとし、処理を行い、結果を出力データファイル52、または、ローカル入力データファイル62へ出力する。出力をローカル出力データファイル62へ行った場合には、実行ジョブ14の完了後、ジョブ受付実行手段13がファイル転送などによりローカル出力データファイル62を出力データファイル52へコピーする。ジョブ受付実行手段13は実行ジョブ14がローカル入力データファイル1、ローカル出力データファイル2を用いて処理を行った場合、最後にこれらのファイルを消去し、ジョブ実行の処理を完了する。

【0016】ジョブ受付実行手段13は実行ジョブ14の開始時、完了時に、ジョブ実行開始、ジョブ実行完了をジョブ管理手段11に通知する。ジョブ管理手段11は通知によりジョブアイコンの表示を変える。この処理を図8に示す。ジョブ管理手段11はジョブ実行開始通知を受け取ると直ちにジョブファイル選択・移動アイコン212をジョブ実行開始アイコン213に変え、ジョブ実行完了通知を受け取ると直ちにジョブ実行開始アイコン213をジョブ実行完了アイコン218に変える。また、実行ジョブ14の進捗状況を把握できる場合、進捗状況により、ジョブ実行開始アイコン213とジョブ

8

実行完了アイコン218の間で、図9のジョブ実行中アイコン1→4(214→215→216→217)のように、処理量の変化をジョブ実行開始アイコン213がジョブ実行完了アイコン218に段階的に変化していくようにアイコンを変える。これらのアイコン表示処理により、ユーザにジョブの状態を視覚的に把握させることができとなる。

【0017】一方、プロセッサ、ネットワークの表示もジョブを実行中であることを示すよう、アイコンを変化させる。ジョブ管理手段11はジョブ受付実行手段13から受け取ったジョブ実行開始通知をプロセッサ管理手段12に送る。通知を受け取るとプロセッサ管理手段12はジョブ実行中のサーバ計算機を表すアイコンを変更、または、ジョブを表すアイコンをサーバ計算機のアイコンとともに表示する。図10にこの例を示す。プロセッサ1がジョブ1を実行している場合、もとのCPU31をジョブ実行中プロセッサアイコン311、または、ジョブ実行中プロセッサアイコン312に変える。ジョブの実行完了時には、変更したアイコンをもとに戻す。

どのジョブを実行中であるかがわかるように、ジョブファイルのファイル名などを表示した方がよいが、プロセッサ、ネットワークの表示では、接続プロセッサ数が多いとアイコンが小さくなり、文字表示ができなくなることがある。このような場合には、図11に示すように、ジョブを実行しているかしていないかを示すためにアイコンの形や色、パターンのみを変えたものをジョブ実行中プロセッサアイコン(小)313として表示する。ユーザがマウスを操作し、マウスカーソルがジョブ実行中プロセッサアイコン(小)313を指した時にボップアップ形式でジョブ状況表示ウインドウ314を表示する。なお、ジョブ状況表示ウインドウ314にはジョブ名以外の情報も表示されるため、ジョブ実行中プロセッサアイコン311、または、ジョブ実行中プロセッサアイコン312を表示したときも利用可能である。

【0018】以上の処理により、処理実行に関する操作を対話的に扱う機能、および、処理実行の状況を視覚的にユーザに提供する表示機能、を有する分散処理システムを構築できる。

【0019】次に、上記システムに付加すると、より使いやすいシステムに発展させることのできる機能について説明する。

【0020】図12は複数のジョブアイコンを同時に選択する方法を示している。図2ではマウスのボタンを押したままマウスを移動させると、選択したジョブアイコンがマウスとともに移動する機能を示したが、ここではボタンと異なるボタンを利用する。図2の場合とは異なるボタンを押したままマウスを移動させると、領域描画動作となり、ボタンを押した最初の位置とマウスカーソル10のある位置とを対角とする矩形を描く(図12)。矩形領域219の中にアイコンが完全に包含され

たものを選択されたジョブアイコンとし、ジョブ管理手段11はアイコンの表示をジョブ選択アイコン21に変える。領域描画動作を利用しないときは、次のような方法となる。マウスのボタンをジョブアイコン21のところで押し、移動させずにすぐ離すとジョブアイコン21が選択され、続けて異なるジョブアイコン、例えば、ジョブアイコン22のところで同様の操作を行うと、ジョブアイコン22も選択され、結果として、ジョブアイコン21、ジョブアイコン22の二つが選択されることになる。この操作は2回以上の繰り返しが可能である。また、一度選択したジョブアイコンのところで再度マウスのボタンを押すと選択が解除され、ジョブアイコンはもとの状態、すなわち、選択されていない状態にどる。領域描画動作による選択方法と、ボタン押下により順次選択していく方法は、共存可能で、組み合わせて複数のジョブアイコンを選択、または、選択解除を行える。

【0021】図13は複数のプロセッサを同時に選択する方法を示している。ジョブアイコンを複数同時に選択する方法と同様の操作で、実行対象プロセッサをネットワークの中から選択する。ただし、ジョブアイコンを複数同時に選択する方法と異なる点があり、領域描画動作でアイコンを囲んでいく際、ネットワークアーキテクチャだけはアイコンが完全に矩形領域21の内のうちにあっても選択対象としない。これは、プロセッサ単位で複数選択を行っているため、ネットワークを含める限り領域描画動作による選択方法に意味がなくなってしまったのである。このプロセッサ選択でも、領域描画動作による選択方法と、ボタン押下により順次選択していく方法は、共存可能であり、組み合わせて複数のプロセッサアイコンの選択、選択解除を行うことができる。

【0022】次に、一つのジョブアイコンの中に複数の処理単位が記述されている場合の処理を説明する。図14のアイコン21のファイルにはプログラム名、出入力ファイル名等が記述されている(221)。これにはラベルとしてジョブ名を付加し、ジョブのファイル222のよう〔〕で囲んでおく。ジョブ管理手段は〔〕で囲まれた記述を一つの処理単位として扱う。これにより、ファイル223のように〔〕で囲んだ処理単位を複数記述することが可能となる。この場合、ジョブ管理手段は個々の処理単位を独立なものとして扱い、かつ、一つのアイコンに複数の処理単位が含まれていることをユーザーに示すためにアイコンのバターンを変える。図14のジョブアイコン23の場合には、アイコンの左上を変更したものとなっている。

【0023】ジョブを実行するときに選択したプロセッサが、マルチプロセッサや並列計算機のように、複数の独立動作可能なCPUを併設している場合、プロセッサを指定したときの処理を図15を用いて説明する。ジョブ移動アイコン25を複数のCPUを登録するプロセッ

サのアイコンに移動させ、マウスのボタンを離すと、プロセッサオプションシート50が開き、アイコンのジョブを並列処理、CPU独立利用のどちらのモードで実行するかを、ユーザーに問い合わせる。並列処理の場合は並列処理選択ボタン510を、CPU独立利用の場合は独立利用選択ボタン520を選択する。並列処理を選択する場合には、実行CPU数指定フィールド511でCPU数を指定する。CPU数は計算機システムがユーザーに許可しているプロセッサ数の最大値を越えて指定することはできない。また実行CPU数指定フィールド511にはデフォルト値としてユーザーに許可しているプロセッサ数の最大値が設定される。この値は一般に並列計算機の運用上、システム管理者によって設定されている。独立利用を選択すると、さらにボップアップ形式で実行CPU選択シート521が表示され、これにユーザーが利用可能なCPUのリストが表示される。ユーザーは実行CPU選択シート521から、ジョブを実行させるCPUを選択する。CPUの選択操作が完了すると、実行CPU選択シート521は閉じられる。リストの中からAN20 Yの項目を選択した場合、実際にジョブを実行するCPUは並列計算機システムにより決定される。また、プロセッサ管理手段12の環境設定により、実行CPU選択シート521に表示されるCPUのリストをアイコンで表示することができ、このモードでは実行CPU選択シート521は一覧なく開いた状態のままになり、ウインドウクローズの指定をするまで閉じられない。各CPUアイコンはプロセッサ管理手段12に表示されるプロセッサアイコンと同じ扱いとなる。したがって、プロセッサアイコンに対して行う操作と同じ操作でCPUを選択、または、特定することができる。

【0024】プロセッサ管理手段12で表示されるプロセッサアイコンにプロセッサの状態を表示情報を付加すると、ユーザーのプロセッサ指定の操作をより使いやすいものにすることができる。この機能を図16を用いて説明する。プロセッサ管理手段12はジョブ実行受付手段13より送られてくるジョブ実行受付手段13が動いているプロセッサのCPU稼働率、ログインユーザーなどから計算機システムとしての状態をランク分けし、それぞれを異なるバーン、色などでプロセッサアイコンを表示する。図16は五つのCPU31～35がネットワークアイコン30に接続されており、このうち、CPU32にかなり負荷がかかっている状態を表している。また、状態のランク分けを定的に示す識別部530が同じウィンドウ内に表示されている。これにより、ユーザーが実行プロセッサを選択する際の判断材料を視覚的に提供することができる。計算機システムは一般に絶えず変動しているため、状態が細かくランク分けしてもすぐに変化してしまうため、あまり効果はない。また、細かくランク分けると、その管理、表示などの処理量が多くなり、ユーザー

11

ジョブ実行の妨げになることもあり得る。ランク分けは3~15段階程度とし、ランク分け数、および、プロセッサ状態ランク識別部530の表示、非表示を、ユーザが設定、選択可能とする。また、状態チェックの時間間隔は短ければ短いほどよいが、短くなるにつれユーザジョブ実行の妨げになる場合があるため、プロセッサ管理手段12のパラメータとしてユーザが設定できるようにしておく。デフォルト値はプロセッサ管理手段12のセットアップのときに、システム管理者が行う。

【0025】次に、本発明によるプロセッサ指定方式を応用し、一つのプログラムをプロック単位でバイブライン的に動作させるための実行指示例を図17に示す。実行可能プログラムR 61、62、63は実行プログラム一覧ウインドウ60にアイコンで表示され、選択・表示操作を行うと、別ウインドウ70にその内部構造がアイコンイメージで表示される。図17は61を選択・表示された例を示しており、61は三つのプロック(BLK)71、72、73、および入力データ指定部74、出力データ指定部75より構成されている。入力データ指定部74を通して与えられたデータにより、まず、71が処理を行い、その結果を72へ送り、72が処理を行う。同様に、72は処理結果を73へ送り、73は受け取ったデータで処理を行った結果を61の出力として出力データ指定部75へ送出する。まず、ユーザは、処理プロックのアイコンを図1に示した図3と同様な操作で各プロックをそれぞれ異なるプロセッサに割当てる。例えば71をマスクを用いてCPU31のところへ移動させると、BLK 71はCPU31で処理されることが定義される。同様に、BLK 72をCPU32に、BLK 73をCPU33に割当て、実行ジョブウインドウ1内にアイコンで表示されているジョブアイコン21を入力データ指定部74に移動させると、ジョブアイコン21をジョブ実行指示として入力データ指定部74が認識し、プログラムPRの実行を行う。他のデータを統けて、または複数のジョブファイルを選択して同時に入力データ指定部74に与えると、入力データ指定部74は順次BLK 71にデータを送り、ジョブ実行の処理を行。ジョブはBLK 71、BLK 72、BLK 73により、複数の入力データに対してバイブルайн的に実行される。特に、共有メモリ方式の並列計算機を用い、各プロックの入出力データを各プロセッサから共通にアクセス可能な領域に保持しておくと、データ転送の必要がなくなるため、プロック間の転送データ量が多くても高速処理が可能となる。

【0026】

【発明の効果】本発明では、複数のジョブをネットワークで接続された計算機システムを用いて分散処理を行う際に、個々のジョブを複数のプロセッサに割り当てる操作を対話的に行え、また、ユーザが定義しなかった場合にはジョブ管理手段が複数を割り当てを設定する自動処

12

理を有することにより、ユーザがジョブの実行状態を視覚的に捉えることが可能となる。ジョブを実行するプロセッサはその状態も含めてアイコンなどのグラフィックイメージで表示することにより、選択すべきプロセッサを容易に把握することができる。

【0027】また、本発明のプロセッサ指定方式は單一プログラムをブロック毎の処理に分解した場合にも適用でき、この場合複数の入力データをバイブルайн的に順次処理するような応用が可能である。入力データファイルの数が多いほど、バイブルайн動作による高速化の効果が得られる。

【図面の簡単な説明】

【図1】本発明によるプロセッサ指定方式の基本方式を示すロック図。

【図2】本発明によるプロセッサ指定方式のジョブアイコンを選択する方式を示す説明図。

【図3】本発明によるプロセッサ指定方式のプロセッサアイコンを選択する方式を示す説明図。

【図4】本発明によるプロセッサ指定方式のネットワークアイコンを選択する方式を示す説明図。

【図5】本発明によるプロセッサ指定方式の複数なサーバ計算機を選択する手順の説明図。

【図6】本発明によるプロセッサ指定方式のシステムがアイコンを移動する機能の説明図。

【図7】本発明によるプロセッサ指定方式におけるジョブの実行方法の説明図。

【図8】本発明によるプロセッサ指定方式のジョブ実行中のジョブアイコン制御を示す説明図。

【図9】本発明によるプロセッサ指定方式のジョブ実行中のジョブアイコン制御を示す説明図。

【図10】本発明によるプロセッサ指定方式のジョブ実行中のプロセッサアイコン制御を示す説明図。

【図11】本発明によるプロセッサ指定方式のジョブ実行中のプロセッサアイコン制御を示す説明図。

【図12】本発明によるプロセッサ指定方式のジョブアイコンの複数選択を示す説明図。

【図13】本発明によるプロセッサ指定方式のプロセッサアイコンの複数選択を示す説明図。

【図14】本発明によるプロセッサ指定方式においてシヨブファイルに記述されている処理単位の数の扱いの説明図。

【図15】本発明によるプロセッサ指定方式において複数のCPUを含むプロセッサを選択する方法の説明図。

【図16】本発明によるプロセッサ指定方式のプロセッサの状態表示の説明図。

【図17】本発明によるプロセッサ指定方式の応用例の説明図。

【符号の説明】

1…ユーザ計算機、2…サーバ計算機、3…ネットワーク、11、13…ジョブ受付実行手段、12…プロセッ

13

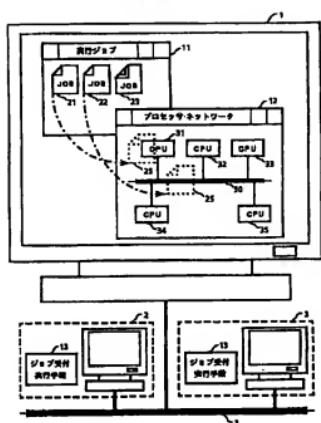
サ管理手段、21、22、23…ジョブアイコン、25
…ジョブ移動アイコン、30…ネットワークアイコン、

14

31、32、33、34、35…CPU。

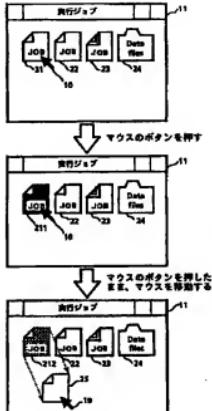
【図1】

図 1



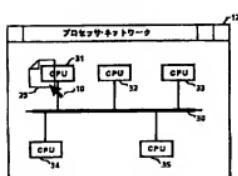
【図2】

図 2



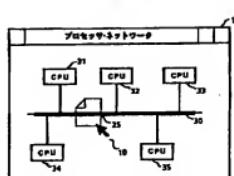
【図3】

図 3

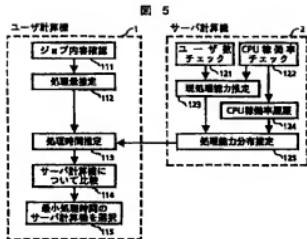


【図4】

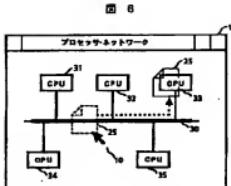
図 4



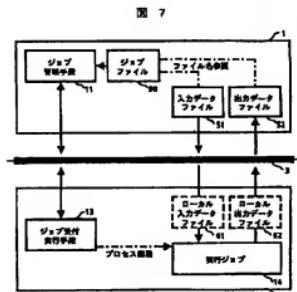
【図5】



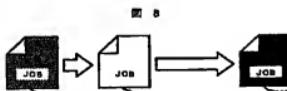
【図6】



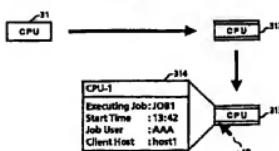
【図7】



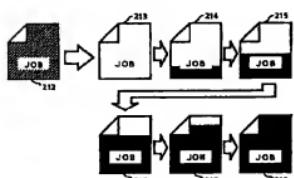
【図8】



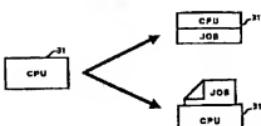
【図11】



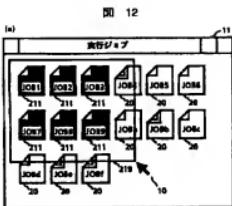
【図9】



【図10】



[图 1-2]



【图 1-3】

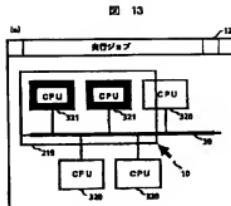
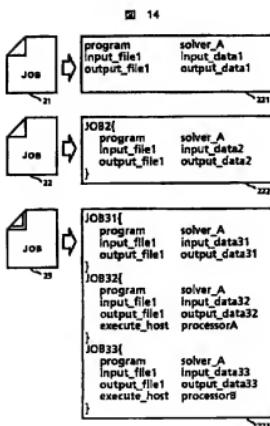


圖 12

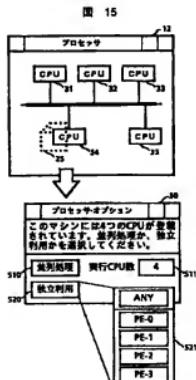
A screenshot of the Windows Task Manager. The title bar says '実行プログラム' (Running Programs). Below the title bar, there is a list of processes. The first four processes are identical, showing 'SHELL32.dll' as the application name, '0x00000000' as the CPU usage, '0x00000000' as the memory usage, and '0x00000000' as the priority. The fifth process is also 'SHELL32.dll' but with different values: '0x00000000' for CPU, '0x00000000' for memory, and '0x00000000' for priority.

The diagram illustrates the internal structure of a RISC-V CPU core. It features a central **CPU** block connected to various functional units: a **ALU**, a **MULT**, a **MEM** (Memory), and a **PC** (Program Counter). The **ALU** and **MULT** units have bidirectional connections between them. The **PC** is connected to the **ALU** and **MULT**. The **MEM** unit is connected to both the **ALU** and **MULT**. The **CPU** block also has connections to an **IMM** (Immediate Value) register, a **REGFILE** (Register File), and a **PC** (Program Counter). The **REGFILE** is connected to the **ALU**, **MULT**, and **MEM** units. The **PC** is connected to the **REGFILE** and the **ALU**.

[图 1-1]

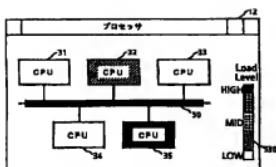


[图 1-5]



【図16】

図 16



【図17】

図 17

